Article

OPT-FRAC-CHN: Optimal Fractional Continuous Hopfield Network

Karim El Moutaouakil 1,*,1, Zakaria Bouhnach 1,†, Abdellah Ahourag 1, Ahmed Aberqi 2 and Touria Karite 3

1 Laboratory of Engineering Sciences, Multidisciplinary Faculty of Taza, Sidi Mohamed Ben Abdellah University, Taza 35000, Morocco; zakaria.bouhnach@usmba.ac.ma (Z.B.); abdellah.ahourag@usmba.ac.ma (A.A.)
2 Laboratory of Mathematics Analysis and Applications (LAMA), National School of Applied Sciences, Sidi Mohammed Ben Abdellah University, Fez 30022, Morocco; ahmed.aberqi@usmba.ac.ma
3 Laboratory of Engineering systems and applications (LISA), National School of Applied Sciences, Sidi Mohammed Ben Abdellah University, Fez 30022, Morocco; touria.karite@usmba.ac.ma
* Correspondence: karim.elmoutaouakil@usmba.ac.ma
† These authors contributed equally to this work.

Abstract: The continuous Hopfield network (CHN) is a common recurrent neural network. The CHN tool can be used to solve a number of ranking and optimization problems, where the equilibrium states of the ordinary differential equation (ODE) related to the CHN give the solution to any given problem. Because of the non-local characteristic of the “infinite memory” effect, fractional-order (FO) systems have been proved to describe more accurately the behavior of real dynamical systems, compared to the model’s ODE. In this paper, a fractional-order variant of a Hopfield neural network is introduced to solve a Quadratic Knap Sac Problem (QKSP), namely the fractional CHN (FRAC-CHN). Firstly, the system is integrated with the quadratic method for fractional-order equations whose trajectories have shown erratic paths and jumps to other basin attractions. To avoid these drawbacks, a new algorithm for obtaining an equilibrium point for a CHN is introduced in this paper, namely the optimal fractional CHN (OPT-FRAC-CHN). This is a variable time-step method that converges to a good local minima in just a few iterations. Compared with the non-variable time-stepping CHN method, the optimal time-stepping CHN method (OPT-CHN) and the FRAC-CHN method, the OPT-FRAC-CHN method, produce the best local minima for random CHN instances and for the optimal feeding problem.

Keywords: fractional calculus (FC); continuous Hopfield network (CHN); equilibrium point (EP); quadratic knap sac problem (QKSP)

1. Introduction

A Hopfield network is a spin-glass device designed to simulate neural networks, developed based on Ernst Ising and Wilhelm Lenz’s study of the Ising pattern of magnetic substances [1]. Numerous hard-constrained engineering challenges in a variety of fields have been expressed in terms of Hopfield energy functions: associative memory systems [2], analog-to-digital conversion [3], the job-shop scheduling problem [4], quadratic assignment and other related NP-complete problems [5,6], the channel allocation problem in wireless networks [7], the mobile ad hoc network routing problem [8], image restoration [9], system identification [10], portfolio managing [11,12], classification problems [13,14], and the clustering problem [15,16].

This paper introduces a new version of the Hopfield recurrent neural network, called the optimal fractional continuous Hopfield network (OPT-FRAC-CHN).

The continuous Hopfield network (CHN) is a form of recurrent artificial neural network introduced by John Hopfield in 1982 [17]. It is composed of n neurons that are fully connected and serves as an associative memory system with continuous activation function units; see Figure 1.
The dynamic of this neural network is governed by the following systems of differential Equation (1):

\[
\frac{du}{dt} = Tv + I
\]

where \( T \in \mathbb{R}^{n \times n} \) represents the matrix of the connection weight between the neurons, \( I \in \mathbb{R}^n \) is the vector of the biases, \( u \) is the vector of the states of the neurons, and \( v \) is the vector of outputs of the neurons. The output \( v_i \) of the neuron \( i \) is deduced from the state \( u_i \) by the activation function \( g(\cdot) \) (the hyperbolic tangent). The dynamic of the CHN is associated with an energy function (Lyapunov function) decreases until reaching an equilibrium point. If the matrix \( T \) is symmetric, then the CHN has an equilibrium point \([17]\).

The CHN will solve optimization problems which can be expressed as the constrained minimization of

\[
E_{\text{Lyap}}(v) = -\frac{1}{2}v^T T v - I^T v
\]

The minimums of \( E \) are in the corners of \([0,1]^n\); see \([17–21]\).

To solve a Quadratic Knapsack Problem (QKSP) optimization problem using a CHN, it is necessary to build a suitable energy function obtained by aggregating of all the QKSP components (the objective function and the constraints) using a penalty parameter to control the objective function and the constraints \([22–24]\). The local minima of this energy function correspond to the optimal solutions of the problem to be solved \([25,26]\).

Several researchers have investigated constrained optimization problems in the context of this neural approach \([17–21,27]\). In these papers, however, the reliability of the equilibrium point obtained for the optimization problem is not necessarily assured. Moreover, when constructing the energy function, how the objective function and constraints are taken into account remains a major challenge. A synthetic study of the internal work and of the energy functions introduced to solve the QKSP problem enabled the authors of \([28]\) to...
suggest a general equation that allows QKSP-type problems to be solved via a CHN while ensuring the realism of the equilibrium points. To guarantee the feasibility of the CHN equilibrium point, these authors decomposed the solution space into suitable subsets and guided the convergence of the CHN to a feasible solution. Subsequently, several authors have used this energy function to solve QKSP-type problems [5,15]. In this work, to solve this system using the Euler method, each equation was discretized using a time step [29]. However, this method has a large local truncating error, thereby requiring the use of a very small step size to the detriment of a lower speed and a more rounded error in the floating point computations. Moreover, it is sensitive to scale modification. Nevertheless, this procedure is very vulnerable to initial states and is very time-consuming. To overcome these drawbacks, a variable time-step method whose property is to reduce convergence time has been introduced in [30]. Many authors have already used this method to solve constrained optimization problems [13,16]. However, these papers describe the dynamics of a CHN based on the ordinary derivative, which does not consider the memory effect and prevents a CHN from dealing with interesting details of the optimization problem studied due to insufficient complexity. As it is known, fractional-order systems are a generalization of classical integer-order systems; they incorporate derivatives and integrals of non-integer orders [31,32]. They have gained significant attention recently in various fields due to their ability to model and accurately explain complex and real-world phenomena [14,33]. Their key aspect is that they can capture memory and hereditary properties of processes, which leads to an accurate representation. In this sense, combining fractional-order systems with continuous neural networks can result in a powerful framework for solving complex optimization problems. The authors of [27] used a Hopfield network model with fractional-order neurons for parameter estimation; they used a domain decomposition method to solve the problem. However, an appropriate decomposition remains a challenge, and a poor choice of the three terms of such a decomposition can lead to very poor local minima. In addition, the time step is chosen manually, which can lead to very high processing time and sensitivity.

To improve the capacity of the dynamic equation (1) to describe more accurately the behavior of dynamical systems associated with QKSP, we used the fractional order (FO) in this paper to characterize the infinite memory effect, namely the fractional CHN (FRAC-CHN) [32,33]. To avoid the problem encountered with the decomposition method, the FRAC-CHN system was integrated with the quadratic method for fractional-order equations [34]. To select the best time step, in each iteration, a new algorithm that obtains one equilibrium point for the CHN was introduced in this paper, namely the optimal fractional CHN (OPT-FRAC-CHN). This is a variable time-step method that allows the convergence to a promising local minimum in just a few iterations. We have used this method to find the equilibrium points of random instances of a CHN and to solve the optimal diet problem by building a suitable energy function. Compared to a CHN with a non-variable time step, a CHN with an optimal time step (OPT-CHN)[30], FRAC-CHN, and OPT-FRAC-CHN produce a best local minima of random CHN instances and optimal diet problem (minimum glycemic load, and minimum positive and negative requirements for the nutrients gap). This can be explained by the increased memory of CHNs thanks to the fractional description of the evolution of this recursive neural network. This memory is extended thanks to the quadratic scheme, presented in Section 3.2, which updates the future state using the expertise of four past steps. In addition, OPT-FRAC-CHN gets closer to the global minimum thanks to the optimal selection of time steps, enabling the energy function to be optimally reduced at each iteration.

The main contribution of this work is summarized as follows:

- Introduction of the fractional version of the continuous Hopfield network (CHN);
- Memory augmentation of the CHN by using quadratic fractional numerical schema with a domain local truncating error;
- Introduction of an optimal time step algorithm to solve the fractional CHN model differential equation;
• Solving the optimal regime problem using the FRAC-CHN neural network, the quadratic fractional schema, and the optimal time step algorithm to solve the optimal diet problem.

The rest of the document is organized as follows: Section 2 presents the methodology adopted to realize this work. Section 3 introduces fractional calculus. Section 4 presents the proposed the OPT-FRAC-CHN. Section 5 gives the experimental results. Section 6 provides some conclusions and perspectives.

2. Methodology

In this section, we present the methodology used in conducting this study, as shown in Figure 2. Furthermore, we outline the key notations employed in this paper.

- **Fractional calculus basics:** First, we define the fractional derivative in the Riemann–Liouville sense [32]. Next, we present the quadratic scheme for fractional derivative approximation [34]. Finally, we provide the estimation of the approximation error.

- **Optimal fractional continuous Hopfield network:** First, we give the fractional state differential equation that generalizes the ordinary model given by Equation (1) introduced in [17–21]. Second, we use the quadratic scheme to approximate the fractional derivative of the proposed model given by Equation (17). Third, to ensure a maximal decrease of the fractional energy function, we calculate the optimal step using an explicit equation; see Equation (13). Then, we give the approximation error majoration in the case of the fractional CHN; see Equations (11) and (17). Fourth, we build a suitable energy function, see Equation (14), to solve the optimal diet problem using the optimal fractional recurrent neural network [5,15]. In this regard, the objective function and the constraints of the mathematical model (see Equation (14)), introduced in [35–40], are combined using penalty parameters to control the feasibility and optimality of the resulted regime. Fifth, we give the algorithms of the algorithms associated with the CHN (Algorithm 1), OPT-CHN (Algorithm 2), FRAC-CHN (Algorithm 3), and OPT-FRAC-CHN (Algorithm 4).

- **Experimentation and implementation details:** In this stage, the properties of the decreasing energy function and convergence have been verified by a first series of computational experiments based on 100 random CHNs of different sizes [30]. In addition, we use the OPT-FRAC-CHN to solve the optimal feeding problem [35–37]:

  - Constraints and objective function parameters are extracted from a set of 177 Moroccan foods described based on 20 nutrients;
- Positive and negative nutrient requirements are extracted from the recommendations of the World Health Organization (WHO) and the Food and Agriculture Organization of the United Nations (FAO) [38–40].

**Algorithm 1 CHN algorithm**

Fractional order: $\alpha$;  
Time step: $\delta t$;  
Initial outputs: $v^0, v^1, v^2,$ and $v^3$;  
Generate the initial population  
while $i < MaxIter$ do  
$v^{i+1} = v^i + \delta t \nabla E(v^i)$  
end while  
return Equilibrium point $v^{MaxIter,*}$

**Algorithm 2 OPT-CHN algorithm**

Fractional order: $\alpha$;  
Initial time step: $\delta t^0$;  
Initial outputs: $v^0, v^1, v^2,$ and $v^3$;  
Generate the initial population  
while $i < MaxIter$ do  
$v^{i+1} = v^i + \delta t_i \nabla E(v^i)$  
end while  
return Equilibrium point $v^{MaxIter,*}$

**Algorithm 3 FRAC-CHN algorithm**

Fractional order: $\alpha$;  
Time step: $\delta t$;  
Initial outputs: $v^0, v^1, v^2,$ and $v^3$;  
Generate the initial population  
while $i < MaxIter$ do  
$v^{i+1} = A_i + \delta t_i B_i$  
end while  
return Equilibrium point $v^{MaxIter,*}$

**Algorithm 4 OPT-FRAC-CHN algorithm**

Fractional order: $\alpha$;  
Initial time step: $\delta t^0$;  
Initial outputs: $v^0, v^1, v^2,$ and $v^3$;  
Generate the initial population  
while $i < MaxIter$ do  
$v^{i+1} = A_i + \delta t_i B_i$  
end while  
return Equilibrium point $v^{MaxIter,*}$
3. Fractional Calculus Basics

In this section, we present the Riemann–Liouville fractional integral \([32]\). Then, we give some very easy-to-use quadratic schemes for fractional derivative approximation \([34]\).

3.1. Fractional Derivative

The Riemann–Liouville \([32]\) fractional integral of order \(\alpha > 0\) is defined as

\[
I^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_a^t (t-\tau)^{\alpha-1} f(\tau) d\tau
\]

and \(I^0 f(t) = f(t)\).

The fractional derivative known as the Riemann–Liouville fractional derivative of order \(n - 1 < \alpha \leq n\) is defined as

\[
D^\alpha f(t) = \frac{1}{\Gamma(n-\alpha)} \left( \frac{d}{dt} \right)^n \int_a^t (t-\tau)^{n-\alpha-1} f(\tau) d\tau
\]

where \(n\) is an integer. Another definition of the fractional derivative introduced by Caputo \([33]\), of order \(m - 1 < \alpha \leq m\), is defined as

\[
D^\alpha f(t) = \frac{1}{\Gamma(m-\alpha)} \int_a^t (t-\tau)^{m-\alpha-1} f^{(m)}(\tau) d\tau
\]

where \(m\) is an integer. In the next subsection, we will give a quadratic scheme to approximate the fractional derivative given by Equation (5).

3.2. Quadratic Scheme for Fractional Derivative Approximation

In this subsection, the domain interval \([0, t]\) is distributed into an even number of subintervals, \(N = 2n\) for \(1 \leq n\), in equal parts with a uniform step size (or time interval) \(h\), where \(h = \frac{t}{2n}\) such that the node points are \(t_i = ih; i = 0, 1, 2, ..., 2n\).

\[
I^\alpha f(t) \approx IQ(f, h, \alpha) \text{ and } D^\alpha f(t) \approx DQ(f, h, \alpha) \text{ and } E_{IQ} I^\alpha f(t) = I^\alpha f(t) - IQ(f, h, \alpha)
\]

and

\[
E_{DQ} I^\alpha f(t) = D^\alpha f(t) - DQ(f, h, \alpha)
\]

where \(IQ(f, h, \alpha)\) and \(DQ(f, h, \alpha)\) are the quadratic approximations of \(I^\alpha f(t)\) and \(D^\alpha f(t)\), respectively. \(E_{IQ} I^\alpha f(t)\) and \(E_{DQ} I^\alpha f(t)\) are the errors associated with these approximations, respectively.

The function \(f(\tau)\) is approximated over the interval \([t_2, t_{2i+2}]\) using the following formula \([34]\):

\[
f_{2i} = \frac{-(\tau - t_{2i+1})}{2h} [1 - \frac{\tau - t_{2i+1}}{h}] f_{2i} + [1 - \left( \frac{\tau - t_{2i+1}}{h} \right)^2] f_{2i+1} + \frac{\tau - t_{2i+1}}{2h} [1 + \frac{\tau + t_{2i+1}}{h}] f_{2i+2}
\]

We suppose that \(f \in C^{m+3}[0, \delta]\) and that the interval \([0, \delta]\) is divided into an even number of subintervals \([t_2, t_{2i+2}]\) such that \(t_i = ih\) with \(h = \frac{\delta}{2n}\) \(i = 0, 1, 2, ..., 2n\); then, the quadratic approximation \(DQ(f, h, \alpha)\) of the D-operator is given by \([41]\)

\[
DQ(f, h, \alpha) = \sum_{i=0}^{n} (A_m f^{(m)}(t_{2i}) + B_m f^{(m)}(t_{2i+1}) + C_m f^{(m)}(t_{2i+2}))
\]

where
We define the following function:

Theorem 1. The approximation error has the form

\[ |E_{DO}(f, h, \alpha)| \leq C_\alpha \| f^{(m+3)} \|_{\infty} T^{(m-\alpha)} h^3 \]  

(8)

where \( C_\alpha \) is a constant that solely depends on \( \alpha \) \([41]\).

4. Optimal Fractional Continuous Hopfield Network

In this section, we give the fractional continuous Hopfield network that implements the outputs of the neurons only. Then, we demonstrate that this equation has a unique solution. In addition, we introduce our method called the OPT-FRAC-CHN. Finally, we build a specific energy function to solve the optimal diet problem.

4.1. Fractional Continuous Hopfield Network

The dynamical system (1) can also be expressed as

\[
\begin{aligned}
\frac{dv_i}{dt} &= \frac{2}{n_0} v_i (1 - v_i) (\sum_{j=1}^n T_{ij} v_j + I_i), \forall i = 1, \ldots, n \\
v_i(0) &= v_i^0 \in [0, 1], \forall i = 1, \ldots, n
\end{aligned}
\]  

(9)

The dynamic of the FRAC-CHN is governed by

\[
\begin{aligned}
\frac{dv_i}{dt} &= \frac{2}{n_0} v_i (1 - v_i) (\sum_{j=1}^n T_{ij} v_j + I_i), \forall i = 1, \ldots, n \\
v_i(0) &= v_i^0 \in [0, 1], \forall i = 1, \ldots, n
\end{aligned}
\]  

(10)

We define the following function:

\[ F(t, v) = \frac{2}{n_0} v_i (1 - v_i) (\sum_{j=1}^n T_{ij} v_j + I_i), \quad \forall i = 1, \ldots, n \]

Theorem 2. The system of Equation (10) has a unique solution.

Proof. For the system (10), to find a unique solution, we must demonstrate that the function \( F \) is Lipschitizn concerning \( v \). \( |F(t, v) - F(t, w)| = \frac{2}{n_0} |v_i (1 - v_i) (\sum_{j=1}^n T_{ij} v_j + I_i) - w_i (1 - w_i) (\sum_{j=1}^n T_{ij} w_j + I_j)| \)

By adding and subtracting the \( v_i (1 - v_i) (\sum_{j=1}^n T_{ij} v_j + I_i) \) term, we obtain

\[ |F(t, v) - F(t, w)| = \frac{2}{n_0} |v_i (1 - v_i) (\sum_{j=1}^n T_{ij} v_j + I_i - (1 - v_i) (\sum_{j=1}^n T_{ij} w_j + I_j)) + v_i (1 - v_i) (\sum_{j=1}^n T_{ij} v_j + I_i) - w_i (1 - w_i) (\sum_{j=1}^n T_{ij} w_j + I_j)| \]

Using the triangle inequality and knowing that \( v_i \in [0, 1] \) (thus \( |v_i (1 - v_i)| = v_i (1 - v_i) \)), we obtain the following inequality:

\[ |F(t, v) - F(t, w)| \leq \frac{2}{n_0} |v_i (1 - v_i) (\sum_{j=1}^n T_{ij} v_j - \sum_{j=1}^n T_{ij} w_j)| + \frac{2}{n_0} |\sum_{j=1}^n T_{ij} v_j + I_i - w_i (1 - w_i) (\sum_{j=1}^n T_{ij} w_j + I_j)| \]

As \( v_i \in [0, 1], \forall i = 1, \ldots, n \), we have \( |\sum_{j=1}^n T_{ij} v_j + I_i| \leq \sum_{j=1}^n |T_{ij}| + |I_i| \), and \( |\sum_{j=1}^n T_{ij} w_j + I_j| \leq \sum_{j=1}^n |T_{ij}| + |I_j| \), \( \forall i = 1, \ldots, n \). We set \( S_1 = \max \{ \sum_{j=1}^n |T_{ij}| + |I_i|, i = 1, \ldots, n \} \) and \( S_2 = \max \{ \sum_{j=1}^n |T_{ij}|, i = 1, \ldots, n \} \); as we know that \( \max \{ a(1 - a), a \in [0, 1] \} = \frac{1}{4} \), we get the following inequality:

\[ |F(t, v) - F(t, w)| \leq \frac{2S_1}{n_0} \| v - w \|_\infty + \frac{2S_2}{n_0} |v_i (1 - v_i) - w_i (1 - w_i)| \]
We have $|v_i(1 - v_i) - w_i(1 - w_i)| \leq |v_i - w_i|(1 + v_i + w_i) \leq 3|v_i - w_i|$. Then, we get the following result:

$$|F_i(t, v) - F_i(t, w)| \leq \frac{3(2 + 3S_i)}{\Delta t} ||v - w||_{\infty}, \forall i = 1, ..., n. \quad \Box$$

**Lemma 1.** $\forall i = 1, ..., n$ and $\forall d \in \mathbb{N}$, $\frac{d^d v}{dt^d}$ is a polynomial considering the vector $v$.

**Proof.** $\forall i = 1, ..., n$, $\frac{d^d v}{dt^d} = \frac{2}{\Delta t^d} v_i(1 - v_i)(\sum_{j=1}^{n} T_{ij}v_j + I_i)$, thus $\frac{d^d v}{dt^d}$ is a polynomial of degree 3.

We suppose that $\frac{d^d v}{dt^d}$ is a polynomial $p(v)$ of $q$ degree considering the vector $v$:

$$p(v) = \sum_{k=1}^{m} a_k \prod_{i=1}^{n} v_i^{n_{i,k}}, \quad \text{where} \quad \sum_{i=1}^{n} n_{i,k} \leq q, \quad \forall k$$

$$\frac{d^d v}{dt^d} = \frac{d p(v)}{dt} = \sum_{k=1}^{m} a_k \sum_{j=1}^{n} n_{j,k} v_j^{n_{j,k}-1} \prod_{i \neq j} v_i^{n_{i,k}}$$

Thus, $\frac{d^d v}{dt^d}$ is a polynomial of degree $\text{Max}\{2 + \sum_{j=1}^{n} n_{j,k}, k = 1, ..., m\} = 2 + q \quad \Box$

**Theorem 3.** The approximation error has the form

$$\forall i \quad \text{Max}\{|E_{DQ}(v_i, t, h, \alpha)|, i = 1, ..., n\} \leq C_\alpha C_{m,T,t} t^{m-\alpha} h^3$$

where $C_\alpha$ is a constant depending only on $\alpha$ and $C_{m,T,t,u_0}$ is a constant depending only on $m, T, I,$ and $u_0$.

**Proof.** for Theorem 1 and Lemma 1, we have $|E_{DQ}(v_i, t, h, \alpha)| \leq C_\alpha \|v_i^{(m+3)}\|_{\infty} t^{m-\alpha} h^3$, where $C_\alpha$ is a constant depending only on $\alpha$.

As $v_i^{(m+3)}$ is a polynomial on $v$ and $v_i \in [0, 1]$, then $\|v_i^{(m+3)}\|_{\infty} \leq C_{m,T,t,u_0}$, where $C_{m,T,t,u_0}$ is a constant depending only on $m, T, I,$ and $u_0$ (the slope of the neuron activation function).

Let us define $C_\alpha = \text{Max}\{C_\alpha, i = 1, ..., n\}$.

Then, we have the desired result. \quad \Box

### 4.2. Fractional Continuous Hopfield Network with Optimal Time Step

In the OPT-FRAC-CHN, the update process is based on fractional calculus (FC). The derivation in Equation (10) is solved by considering the first four terms of the derivative as

$$v^{i+1} = \alpha v^i + 1/2\alpha v^{i-1} + \frac{1}{6}\alpha(1 - \alpha) v^{i-2} + \frac{1}{24}\alpha(1 - \alpha)(2 - \alpha) v^{i-3} - \delta t \frac{2}{u_0} v^i (1 - v^i) \nabla E(v^i)$$

where $\alpha$ is a constant in the interval $[0, 1]$, $\Delta t$ is the time step, $\delta v$ represents the neuron outputs at iteration, $i$, $v^{i-1}$ is that at iteration $i - 1$, and so on. Figure 3 gives an electronic representation of the discrete fractional continuous Hopfield lattice. In this sense, to calculate $\delta v$, we use $v^{i-1}$, $v^{i-2}$, and $v^{i-3}$ using the appropriate weights obtained from the discrete approximation of the fractional derivative, namely $\alpha$, $1/2\alpha$, $\frac{1}{6}\alpha(1 - \alpha)$, and $\frac{1}{24}\alpha(1 - \alpha)(2 - \alpha)$, respectively.
Figure 3. Electronic diagram of a discrete-time Hopfield lattice of order $\alpha$

To determine the equilibrium point of the suggested recurrent neural network, we present the subsequent algorithm:

**Theorem 4.** Let us define $\alpha$ as a constant in the interval $[0, 1]$ and $v^i, v^{i-1}, v^{i-2},$ and $v^{i-3}$ as the states of the FRAC-CHN at the iteration $i, i-1, i-2,$ and $i-3,$ respectively. Then, the optimal fractional time step is given by

$$\delta t^i = \frac{B'_i(TA_i + I)}{B'_iTB_i}$$

where $B_i = \frac{2}{\hat{m}_0}v^i(1 - v^i)\nabla E(v^i)$ and $A_i = \alpha v^i + 1/2\alpha v^{i-1} + \frac{1}{6}a(1 - \alpha)v^{i-2} + \frac{1}{24}a(1 - \alpha)(2 - \alpha)v^{i-3}.$

**Proof.** First, we set $\phi(\nabla t) = E(\alpha v^i + 1/2\alpha v^{i-1} + \frac{1}{6}a(1 - \alpha)v^{i-2} + \frac{1}{24}a(1 - \alpha)(2 - \alpha)v^{i-3} - \delta t \frac{2}{\hat{m}_0}v^i(1 - v^i)\nabla E(v^i)).$

$dt^{i,*}$ is the solution of

$$\min_{dt \in [-1, +1]} \{ \phi(dt^{i,*}) \} \text{ if } \frac{d\phi}{dt}(dt^{i,*}) = 0$$

Thus, $B'_i\nabla E(A_i - \delta tB_i) = 0.$ Then, $B'_i(T(A_i - dt^{i,*}B_i) + I) = 0.$

Thus,

$$dt^{i,*} = \frac{B'_i(TA_i + I)}{B'_iTB_i}$$
4.3. Application: OPT-FRAC-CHN to Optimal Diet Problem

The fuzzy quadratic optimization programming \((P)\) that minimizes the total glycemic load, and minimizes the favorable and unfavorable nutrient gaps \([42,43]\) is given by Equation \([35–40]\):

\[
(P): \left\{ \begin{array}{l}
\min g'x + \alpha/2\|Ax - b\|^2 + \beta/2\|Ex - f\|^2 \\
\text{Subject to:} \\
x \in \{0,1\}^d
\end{array} \right. 
\]  

(14)

where the terms are defined as follows:

- \(d\) is the number of foods;
- \(x = (x_i)_{i=1}^d\) is the vector of the foods serving sizes;
- \(g\) is the vector formed by the foods’ glycemic load;
- \(A\) is the matrix of the favorable nutrients, \(b\) is the vector of the favorable nutrient requirements, \(E\) is the vector of unfavorable nutrients, and \(f\) is the maximum number of positive nutrients that the diet must contain,
- \(\mu\) and \(\beta\) are penalty parameters that achieve the balance between the objective function components. In practice, if \(\beta \leq \mu\) the CHN will focus on the objective function to the detriment of the constraints; otherwise, the constraints will attract more of the CHN’s attention \([13,16,35,37,38]\). \(u\) is the vector of ones from \(\mathbb{R}^d\).

To solve the problem \((P)\) using the CHN, we introduce the following energy function:

\[
E(x) = g'x + \mu/2\|Ax - b\|^2 + \beta/2\|Ex - f\|^2 + \gamma/2x(1-x)
\]  

(15)

where \(\gamma\) is a penalty parameter that controls the binarity of the output of the CHN units. In this sense, the differential equation that governs the dynamic of the diet is given by

\[
\frac{dv_i}{dt} = \frac{2}{\alpha_0}v_i(1 - v_i)(\sum_{j=1}^n T_{ij} v_j + I^D_i), \forall i = 1, ..., n
\]

\[
v_i(0) = v_i^0 \in [0,1], \forall i = 1, ..., n
\]  

(16)

The dynamics of the FRAC-CHN related to optimal dietary issues are controlled by

\[
\frac{dv_i}{dt} = \frac{2}{\alpha_0}v_i(1 - v_i)(\sum_{j=1}^n T_{ij} v_j + I^D_i), \forall i = 1, ..., n
\]

\[
v_i(0) = v_i^0 \in [0,1], \forall i = 1, ..., n
\]  

(17)

To calculate \(T^D\) and \(I^D\), we calculate the first and the second derivatives of the function (15):

\[
\frac{dE}{dx} = g + \mu A'(Ax - b) + \beta E'(Ex - f) + \gamma/2(1 - 2x)
\]

\[
\frac{d^2E}{dx^2} = \mu A' + \beta E' - \gamma 1_d
\]

The bias and the weight of the built CHN are given by

\[
I^D = -\frac{dE}{dx}(0) = -g + \mu A'b + \beta E'f - \gamma/2
\]

\[
T^D = -\frac{d^2E}{dx^2}(0) = -\mu A' + \beta E' + \gamma 1_d
\]  

(18)

To avoid the stability of the CHN at the interior of \([0,1]^d\), we introduce the following constraint \([15,16]\):

\[
\gamma \geq \mu \max(\text{diag}(A'A)) + \beta \max(\text{diag}(E'E))
\]  

(19)
4.4. Proposed Algorithm

The algorithms associated with the CHN, OPT-CHN, FRAC-CHN, and OPT-FRAC-CHN will be used to find the equilibrium point of different instances. It will therefore be of great educational interest to recall the main instructions of these algorithms.

Algorithm 1 presents the main steps of the basic CHN [5,17–19]. This algorithm builds a trajectory based on the same time step over the course of iterations.

Algorithm 2 presents the main steps of the OPT-CHN [15,26,28]. This algorithm builds a trajectory based on an optimal time step over the course of iterations to correct the fluctuation of the FRAC-CHN. However, this method only takes into account the neuron outputs of the previous iteration $v^i$.

Algorithm 3 presents the main steps of the FRAC-CHN. This algorithm builds a trajectory based on the same time step throughout iterations [34], which causes fluctuation in the trajectory to the equilibrium point. However, this method takes into account the neuron outputs of the four previous iterations $v^i, v^{i-1}, v^{i-2},$ and $v^{i-3}$, which permits the construction and correction of the trajectory to the equilibrium point.

Algorithm 4 presents the main steps of the OPT-FRAC-CHN. This algorithm builds a trajectory based on optimal time steps over the course of iterations to correct the fluctuation of the FRAC-CHN. In addition, this method takes into account the neuron outputs of the four previous iterations $v^i, v^{i-1}, v^{i-2},$ and $v^{i-3}$, which permits the construction and correction of the trajectory to the equilibrium point using its long memory.

5. Experimentation and Implementation Details

To test the algorithm introduced in this paper, some computer programs were designed to calculate the equilibrium points of several randomly generated CHNs and to solve the optimal diet problem; see Section 5.2. Three programs, which were coded in Matlab, computed one equilibrium point of the CHN by the Euler method, and by the Euler method with optimal step. Two other programs calculated a FRAC-CHN equilibrium point using the quadratic method mentioned in Section 3.2, with a fixed time step and a sequence of optimal steps given by Equation (13).

5.1. Testing and Comparison on Random Instances

The properties of energy function decrease and convergence were verified through a first set of computational experiences based on 100 random CHNs of size different sizes. As in the [30] study, the parameters (weights and biases) of CHNs and FRAC-CHNs were randomly generated in the following way:

$$T = U[-10, 0]^d 	imes d, \quad I = U[0, 5]^d, \quad \text{and} \quad u_0 \in [0.02, 2], \quad \text{where} \ U \ \text{is the uniform distribution.}$$

Figures 4–7 present the energy function of the continuous Hopfield network (Algorithm 1), the optimal continuous Hopfield network (Algorithm 2), fractional continuous Hopfield network ($\alpha = 0.7$, Algorithm 3), and optimal fractional continuous Hopfield network (Algorithm 4) vs. iterations, respectively. We remark that the energy functions of different CHNs decrease with iterations, but the OPT-FRAC-CHN reaches the best equilibrium points. Admittedly, the OPT-CHN has escaped the valley that attracted the CHN, thanks to the optimal pitch calculated at each iteration, but the new minimum remains mediocre compared with that obtained by the FRAC-CHN. Compared with the CHN and OPT-CHN, sampling on $\alpha$ has enabled us to find a population of fractions for which the FRAC-CHN performs better than the CHN and OPT-CHN. This can be explained by the increased memory of CHNs thanks to the fractional description of the evolution of this recursive neural network. This memory is extended thanks to the quadratic scheme, presented in Section 3.2, which updates the future state using the expertise of four past steps. In addition, and considering all almost-instances, the FRAC-CHN was attracted by a local minimum of average quality due to the non-variability of the manually chosen time step. In this context, it is very difficult to analyze the CHN input data (weights and biases) of each instance to choose an appropriate time step. Furthermore, a time step may be adequate for one instance but may produce a very bad solution for a new instance. On the contrary,
the OPT-FRAC-CHN gets closer to the global minimum thanks to the optimal selection of time steps, enabling the energy function to be optimally reduced at each iteration. The other advantage, of a practical nature, is that, once initialized, the evolution of the OPT-FRAC-CHN becomes automatic because the optimal step is given by the Formula (13) as a function of current neuron activations, a fraction, weights, and biases.

Figure 4. CHN energy vs. iterations.

Figure 5. OPT-CHN energy vs. iterations.
To highlight the trajectory generated when calculating the equilibrium points of the CHN, OPT-CHN, FRAC-CHN, and OPT-FRAC-CHN, we have displayed the isolines of the energy function of these CHNs.

Figures 8–11 display the isolines of the energy function of an instance of the CHN. Starting from the same initial states, the trajectories of different methods converge to the optimal equilibrium points. In the classic CHN, the optimal step only allows a few stations to be reduced in the trajectory followed by the CHN when searching for the minimum. For certain instances with several local minima, this reduction may be more remarkable.
Figure 8. CHN energy vs. iteration.

Figure 9. OPT-CHN energy vs. iteration.

Figure 10. FRAC-CHN energy vs. iteration for $\alpha = 0.7$
Given the need for four initializations, the FRAC-CHN reaches the best equilibrium point too late; indeed, this version of the CHN generates several intermediate points before reaching the minimum of the energy function. It is possible to generate the three recalled vectors of neuron activations using the CHN, and then we continued the research using the FRAC-CHN, which improved the behavior of the latter. In our case, the optimal choice of time step, using the Formula (13), enabled us to considerably reduce the number of intermediate points in the search trajectory. In fact, by selecting the optimal time step at each iteration, the OPT-FRAC-CHN ensures the maximum reduction in the CHN’s energy consumption, avoiding the exploration of unpromising regions.

5.2. Optimal Diet Using OPT-FRAC-CHN

We used the OPT-FRAC-CHN to solve the optimal diet problem based on the energy function (15) introduced in Section and by considering the parameter constraints Equation (19) to ensure the feasibility of the equilibrium points.

A feasible solution of Equation (19) is \( \mu = 1.25, \beta = 1, \) and \( \gamma = -0.125. \) The maximum number of iterations is 200. The initial state is randomly generated following where \( U \) is the uniform distribution \( U[0\ 1]. \)

In the optimal diet problem, researchers adopt three performance measures [35–37]: glycemic load, positive nutrient requirement gap, and nutrient requirement gap.

Table 1 gives the total glycemic load, positive nutrient gap, and negative nutrient gap of optimal diets obtained by the CHN, OPT-CHN, FRAC-CHN, and OPT-FRAC-CHN for eight fractional orders (0.5, 0.55, 0.56, 0.57, 0.58, 0.59, 0.6, and 0.61). We remark that the OPT-FRAC-CHN produces optimal regimes with a very low glycemic load, low positive nutrient requirement gap, and negative nutrient requirement gap for different fractional orders. Compared to the CHN, OPT-CHN, and FRAC-CHN, the proposed method produces the best diets. In fact, the CHN and OPT-CHN produce zero glycemic load diets, which is great, but these diets have very large discrepancies in terms of positive and negative nutrient requirements. Compared with the diets produced by the CHN and OPT-CHN, and for almost all \( \alpha \) fraction values, the FRAC-CHN produces the best diet that makes a good compromise between the three criteria: glycemic load, positive nutrient requirement gap, and nutrient requirement gap. This can be explained by the increased memory of CHNs thanks to the fractional description of the evolution of this recursive neural network. This memory is extended thanks to the quadratic scheme, which updates the future state using the expertise of four past steps. However, these diets are still bad compared to the ones produced by the OPT-FRAC-CHN thanks to the optimal step. Indeed, the selection of the optimal time step, in each iteration, permits the OPT-FRAC-CHN to find the best equilibrium points of the CHN compared to the FRAC-CHN.
Table 1. Total glycemic load, positive nutrient gap, and negative nutrient gap of optimal diets obtained by the CHN, OPT-CHN, FRAC-CHN, and OPT-FRAC-CHN.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time Step</th>
<th>Fraction Order</th>
<th>Glycemic Load</th>
<th>Positive Gap (Microgram)</th>
<th>Negative Gap (Microgram)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRAC-CHN</td>
<td>0.001</td>
<td>0.5</td>
<td>367.17</td>
<td>4400.03</td>
<td>2261.45</td>
</tr>
<tr>
<td>OPT-FRAC-CHN</td>
<td>Optimal</td>
<td>0.5</td>
<td>23.96</td>
<td>2039.13</td>
<td>609.57</td>
</tr>
<tr>
<td>Frac-CHN</td>
<td>0.001</td>
<td>0.55</td>
<td>415.42</td>
<td>5853.80</td>
<td>2899.12</td>
</tr>
<tr>
<td>OPT-FRAC-CHN</td>
<td>Optimal</td>
<td>0.55</td>
<td>37.65</td>
<td>1190.53</td>
<td>342.13</td>
</tr>
<tr>
<td>FRAC-CHN</td>
<td>0.001</td>
<td>0.56</td>
<td>442.66</td>
<td>6554.35</td>
<td>4837.73</td>
</tr>
<tr>
<td>OPT-FRAC-CHN</td>
<td>Optimal</td>
<td>0.56</td>
<td>42.78</td>
<td>902.58</td>
<td>244.40</td>
</tr>
<tr>
<td>FRAC-CHN</td>
<td>0.001</td>
<td>0.57</td>
<td>1093.55</td>
<td>24404.71</td>
<td>15717.35</td>
</tr>
<tr>
<td>OPT-FRAC-CHN</td>
<td>Optimal</td>
<td>0.57</td>
<td>43.99</td>
<td>864.32</td>
<td>244.17</td>
</tr>
<tr>
<td>FRAC-CHN</td>
<td>0.001</td>
<td>0.58</td>
<td>43.04</td>
<td>3366.09</td>
<td>720.05</td>
</tr>
<tr>
<td>OPT-FRAC-CHN</td>
<td>Optimal</td>
<td>0.58</td>
<td>47.84</td>
<td>732.78</td>
<td>203.43</td>
</tr>
<tr>
<td>FRAC-CHN</td>
<td>0.001</td>
<td>0.59</td>
<td>1642.23</td>
<td>2508.25</td>
<td>1764.04</td>
</tr>
<tr>
<td>OPT-FRAC-CHN</td>
<td>Optimal</td>
<td>0.59</td>
<td>54.82</td>
<td>562.62</td>
<td>148.48</td>
</tr>
<tr>
<td>FRAC-CHN</td>
<td>0.001</td>
<td>0.6</td>
<td>23.92</td>
<td>3385.86</td>
<td>1046.27</td>
</tr>
<tr>
<td>OPT-FRAC-CHN</td>
<td>Optimal</td>
<td>0.6</td>
<td>49.58</td>
<td>459.62</td>
<td>138.06</td>
</tr>
<tr>
<td>FRAC-CHN</td>
<td>0.001</td>
<td>0.61</td>
<td>899.69</td>
<td>2057.15</td>
<td>8098.13</td>
</tr>
<tr>
<td>OPT-FRAC-CHN</td>
<td>Optimal</td>
<td>0.61</td>
<td>73.66</td>
<td>359.97</td>
<td>98.16</td>
</tr>
<tr>
<td>CHN</td>
<td>0.001</td>
<td>Ordinary</td>
<td>0.00</td>
<td>5036.28</td>
<td>1795.06</td>
</tr>
<tr>
<td>OPT-CHN</td>
<td>Optimal</td>
<td>Ordinary</td>
<td>0.00</td>
<td>5036.28</td>
<td>1795.06</td>
</tr>
</tbody>
</table>

To illustrate the behavior of the energy function, given by Equation (15), we give the dynamic of this function vs. time of the CHN (time step = 0.001), OPT-CHN, FRAC-CHN (time step = 0.001, $\alpha = 0.7$), and OPT-FRAC-CHN ($\alpha = 0.7$); see Figures 12–15; the graph distinguishes the model with the best performance. We remark that the CHN and OPT-CHN were attracted by a medium local minima from the beginning until the end of iterations. We remark that the FRAC-CHN has a periodic behavior, which may prevent the system from reaching an equilibrium point because of the non-variability of the time step. This inconvenience was addressed by the OPT-FRAC-CHN thanks to the optimal time steps calculated at each iteration. We tested the OPT-FRAC-CHN for different $\alpha$ values from $[0 0.5 \cup 0.61 1]$ but the regimes obtained were not satisfactory.

Figure 12. The diet CHN energy vs. iteration.
Figure 13. The diet OPT-CHN energy vs. iteration.

Figure 14. The diet FRAC-CHN energy vs. iteration for $\alpha = 0.7$.

Figure 15. The diet OPT-FRAC-CHN energy vs. iteration for $\alpha = 0.7$. 
5.3. Limitations

In the course of our experimental work, we have identified certain limitations which we discuss in the following subsection.

**Memory complexity:** the fact of considering all four vectors of neuron activation, at each iteration, increases the memory used by a factor of $3 \times \text{size} \text{(problem)} \times \text{size} \text{(float)}$.

**Optimal fraction choice:** to find out which of the fraction values can produce good solutions, we discriminated the interval $[0, 1]$ and looked for an equilibrium point for each of these values, which may cause the best model to catch up.

**Initialization of neuron activation vectors:** it is always difficult to choose a good initialization for a single activation vector, and the problem becomes increasingly difficult when four activation vectors are involved, as this has a strong influence on the quality of the equilibrium points produced.

**Stochastic nature of problem:** the parameters of the food model, given by the Equation (14), are stochastic because the nutrient values of 100 g of a given food change according to several factors (temperature, age, speed ...). However, the CHN does not have enough complexity to capture this phenomenon.

6. Conclusions

Due to the non-local characteristic of the “infinite memory” effect, fractional-order (FO) models have been proven to give a more accurate description of the performance of dynamic actual systems than ordinary differential models. A new fractional model of the CHN, namely the FRAC-CHN, that implements the only neuron outputs was introduced in this paper. To calculate one equilibrium point of the FRAC-CHN, a quadratic integration was used. Thanks to its multistep character, this method permits to correct the trajectory to the equilibrium point. To address this shortcoming, a quadratic integration with optimal time step was introduced in this paper, namely the OPT-FRAC-CHN. This method was used to calculate the equilibrium point of several instances of randomly generated CHNs and to solve the optimal diet problem. Compared with the CHN, OPT-CHN, and FRAC-CHN, thanks to it long memory, its multistep character, and the optimal time step, the OPT-FRAC-CHN produced the best equilibrium points (with a very low energy), corrected the fluctuation of the trajectories to equilibrium points, and needed less iterations to reach these equilibrium points. In short, this superiority can be explained by the increased memory of the CHN thanks to the fractional description of the evolution of this network. This memory was enlarged through the quadratic scheme, which updates the coming state using the expertise of four previous states. In another aspect, the OPT-FRAC-CHN moves closer to the global minimum by selecting the optimal time steps, thereby optimally decreasing the energy function at each iteration.

In conducting our experimental phase, we noted several drawbacks (discussed in Section 5.3): (a) the increase in memory complexity, (b) the difficulty of initializing the activation vectors of the four neurons, and (c) the inability of the CHN to capture the stochastic nature of optimization problems. To overcome these limitations, we can (a) decrease the order of the numerical method used to solve Equation (10), (b) use the classical CHN to obtain the necessary initializations, and (c) use fuzzy logic to describe the evolution of the CHN.

The equilibrium point produced by the OPT-FRAC-CHN can be implemented to solve Quadratic Knap Sac Problems from different fields, in particular the optimal diet problem, image restoration, image segmentation, the portfolio management problem, and the channel allocation problem.

Given that the energy function does not increase, the suggested algorithm consistently reached a local minimum, functioning as an intelligent local search algorithm. Therefore, to prevent being drawn towards local minima, a fuzzy adaptation of the OPT-FRAC-CHN could be a viable solution. Furthermore, the OPT-FRAC-CHN can be expanded to address non-quadratic combinatorial optimization problems.
Author Contributions: Conceptualization, K.E.M.; Methodology, K.E.M.; Investigation, A.A. (Abdellah Ahourag), A.A. (Ahmed Aberqi) and T.K.; Resources, T.K.; Writing—original draft, K.E.M.; Writing—review & editing, Z.B. and A.A. (Abdellah Ahourag); Supervision, K.E.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: Data will be made available to authors with permission.

Acknowledgments: (a) This work was supported by the Ministry of National Education, Professional Training, Higher Education and Scientific Research (MENFPESRS), and the Digital Development Agency (DDA) and CNRST of Morocco (Nos. Alkhawarizmi/2020/23); (b) The authors thank the reviewers and appreciate all valuable comments and suggestions, which helped to improve the quality of the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following are the main notations used in this document:

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP</td>
<td>Equilibrium point</td>
</tr>
<tr>
<td>QKSP</td>
<td>Quadratic Knap Sac Problem</td>
</tr>
<tr>
<td>FC</td>
<td>Fractional calculus</td>
</tr>
<tr>
<td>CHN</td>
<td>Continuous Hopfield network</td>
</tr>
<tr>
<td>T, I</td>
<td>Parameters of CHN network</td>
</tr>
<tr>
<td>OPT-CHN</td>
<td>Optimal continuous Hopfield network</td>
</tr>
<tr>
<td>FRAC-CHN</td>
<td>Fractional continuous Hopfield network</td>
</tr>
<tr>
<td>FC</td>
<td>Fractional Calculus</td>
</tr>
<tr>
<td>OPT-FCRAC-CHN</td>
<td>Continuous Hopfield network</td>
</tr>
<tr>
<td>WHO</td>
<td>World Health Organization</td>
</tr>
<tr>
<td>FAO</td>
<td>Food and Agriculture Organization of the United Nations</td>
</tr>
<tr>
<td>A</td>
<td>Column of positive nutrients of 177 foods</td>
</tr>
<tr>
<td>E</td>
<td>Column of negative nutrients of 177 foods</td>
</tr>
<tr>
<td>g</td>
<td>Vector of the glycemic load of 177 foods</td>
</tr>
<tr>
<td>b</td>
<td>Vector of positive nutrient requirements</td>
</tr>
<tr>
<td>f</td>
<td>Vector of negative nutrient requirements</td>
</tr>
</tbody>
</table>

References


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.